APPLICATION FOR UNITED STATES PATENT

in the name of

Nathaniel McCully

of

Adobe Systems Incorporated

for

METHOD FOR ALIGNING TEXT TO BASELINE GRIDS AND TO CJK CHARACTER GRIDS

Fish & Richardson P.C.
2200 Sand Hill Road, Suite 100
Menlo Park, CA 94025
Tel.: (650) 322-5070
Fax: (650) 854-0875

METHOD FOR ALIGNING TEXT TO BASELINE GRIDS AND TO CJK CHARACTER GRIDS

TECHNICAL FIELD

This invention relates to desktop publishing (DTP) technology, and more particularly to forced grid line spacing control in DTP.

BACKGROUND

When performing character typesetting on one page of an electronic document in DTP, a plurality of reference lines known as a grid is usually arranged on the page displayed on the display device, and characters are arranged with this grid as a reference. This grid defines a plurality of lines separated by a fixed interval, and typesetting is performed by sequentially arranging characters within these lines. Therefore, the size of the characters to be arranged within each line is restricted to a specified dimension, and when this specified dimension is exceeded, characters in adjacent lines may overlap. When the height of text consisting of a plurality of characters to be arranged within one line of the grid exceeds that line's specified dimension, so-called "forced grid line spacing" is performed to arrange these characters within an enlarged space that includes that line and an adjacent line. Forced grid line spacing selects the required number of adjacent lines according to character size, etc., defines an enlarged space by taking two or three lines, and arranges characters within that enlarged space.

However, in conventional DTP for electronic typesetting, either a special setting value must be input when performing forced grid line spacing, or it is necessary to perform forced grid line spacing by manually moving characters displayed on a display device. In addition, when text consists of a plurality of characters with differing fonts or dimensions, doing forced grid line spacing becomes very difficult, and requires much time and effort.

SUMMARY

In general, in one aspect, the invention provides methods and apparatus, including computer program products, implementing and using techniques for controlling forced grid line spacing. The height of text consisting of characters to be arranged within a current line

in a grid displayed on a display device is determined. An arrangement region that includes the current line and at least one subsequent line if the height of the text is larger than a specified dimension for the grid is demarcated. A coordination line within the arrangement region according to a selected coordination mode is set and the characters are arranged within the arrangement region while coordinating the plurality of characters with the coordination line.

Implementations can include one or more of the following features. The grid may be a frame grid that is movable to a desired position on a page of an electronic document displayed on the display device in order to arrange data to be typeset on the page, and may have lines including cells. The grid can be a CJK character grid. The specified dimension of the grid may be a font point dimension that is selected when the grid is created by the user on the display device. The coordination mode may include a top coordination mode, a midpoint coordination mode, a baseline coordination mode, and a bottom coordination mode. Each character may have an associated embox and the maximum dimension of the current line may be the dimension of the largest embox associated with the plurality of characters. The embox may vertically and horizontally delimit the point dimensions of each character and can be an essentially square frame surrounding the character glyph.

Potential advantages of the invention may include one or more of the following. The invention may enable completely automated forced grid line spacing, and may make it possible to perform typesetting operations efficiently by computer processing without the necessity of complicated manual operations, even when the text to be typeset includes characters with differing dimensions. In addition, the invention may make it possible to select a wide variety of forced grid line spacing configurations as options, and may make it possible to do typesetting with much fine detail and variety by computer.

The details of one or more implementations of the invention are set forth in the accompanying drawings and the description below. Other features, objects, and advantages of the invention will be apparent from the description and drawings, and from the claims.

DESCRIPTION OF DRAWINGS

FIG. 1 is a schematic block diagram showing a DTP system capable of controlling forced grid line spacing in accordance with one implementation of the invention.

FIG. 2 is a schematic block diagram showing a DTP system capable of controlling forced grid line spacing in accordance with another implementation of the invention.

FIG. 3 is a schematic diagram showing an example of the display on the screen of a display device when performing typesetting in accordance with one implementation of the invention.

FIG. 4 is a flowchart showing the overall processing procedure for performing typesetting in accordance with one implementation of the invention.

FIG. 5 is a flowchart showing the processing flow for automatically performing forced grid line spacing in accordance with one implementation of the invention.

FIG. 6 is a flowchart showing the processing flow when branching to baseline processing based on the decision result in step S17 in FIG. 5.

FIG. 7A is a schematic diagram showing one example a frame provided with a grid based on CJK characters.

FIG. 7B is a schematic diagram showing text consisting of a plurality of CJK characters with differing dimensions to be arranged within a frame.

FIG. 8A is a schematic diagram showing the situation when the initial characters of the text are arranged within a frame.

FIG. 8B is a schematic diagram showing the remaining text to be arranged within a frame.

FIG. 9A-D are schematic diagrams showing the sequence of states when performing forced grid line spacing (taking two lines) in accordance with one implementation of the invention.

FIG. 10 is a schematic diagram showing the state when all of the text is arranged within a frame.

FIG. 11A-C are schematic diagrams for explaining the embox and the ICF box concepts.

Like reference symbols in the various drawings indicate like elements.

DETAILED DESCRIPTION

The following explanation deals mainly with horizontal typesetting, but it should be noted that the forced grid line spacing control technique described below can be applied to both horizontal typesetting and to vertical typesetting.

FIG. 1 shows an electronic typesetting DTP system 1 in accordance with on one implementation of the invention. DTP processing system 1 has a DTP processing control device 2, which can perform electronic typesetting and layout editing and so forth; a display device 3 for displaying the data being processed; an input means 5 for inputting various types of data such as graphics, text, control instructions, etc.; and an output means 6 for outputting processed data. DTP processing control device 2 has typesetting control means 2a, which controls typesetting, layout control means 2b, which controls layout editing, and font file 2c, which stores font information on attributes of the text to be output on a display or printed, etc. In particular, font file 2c includes CJK fonts 8, which are fonts for ideographic characters such as Japanese, Chinese, Korean, etc. In addition, typesetting control means 2a has forced grid line spacing control means 7, which controls forced grid line spacing when arranging a plurality of characters that are being typeset in electronic typesetting within a grid displayed on the display device.

Display device 3 can comprise various types of display devices such as a CRT or an LCD, etc. Input means 5 can include a keyboard, pointing device (mouse, track ball, track point, etc.), scanner, communication interface, etc. In addition, output means 6 can comprise a printer, external memory device, etc. DTP processing control device 2 can comprise a processor such as a CPU, etc., a memory or an electronic typesetting processing program, etc. that includes a line typesetting control routine that is loaded into memory and processed by a processor. DTP processing control device 2 can alternatively be constituted in part or in whole as hardware constituted as firmware.

FIG. 2 is a block diagram showing a DTP system 10 provided with the inventive forced grid line spacing control function, constituted by applying the invention to a general-purpose computer system. System 10 shown in FIG. 2 has a CPU 11, a memory 12, a CRT 13, a hard disk 14, a keyboard 15, a mouse 16, a CD-ROM drive 17, a FD drive 18, and a printer 19. These units are coupled and work together via a bus 9. In electronic typesetting based on invention, a forced grid line spacing control program 7' is, for example, a routine

- 4 -

with other electronic typesetting programs and is, for example, installed in system 10 via CD-ROM drive 17 or FD drive 18, and stored on hard disk 14. Font file 2c can also be stored on hard disk 14. When the forced grid line spacing control program 7' or an electronic typesetting program containing the forced grid line spacing control program is started by a user operating keyboard 15 or mouse 16,the forced grid line spacing control program is loaded into memory 12 as shown in the diagram. Therefore, forced grid line spacing control means 7 be constituted as CPU 11 and forced grid line spacing control program 7' loaded into memory 12. Therefore, in one aspect the invention comprises a computer-readable recording medium (for example, CD-ROM, FD, tape, semiconductor memory, etc.) storing a forced grid line spacing control program so that a general-purpose computer system can be converted into a DTP system providing the inventive forced grid line spacing control function.

FIG. 3 shows part of the screen of the display device 3 in FIG. 1, which can be CRT 13 in FIG. 2, for example. Electronic document 122 is displayed on screen 121 as an editing form, and layout editing and typesetting editing of various page structural elements are performed thereon. Electronic document 122 can be displayed on screen 121 by clicking with a mouse on "File", in menu bar 120 that is displayed along the top of screen 121, and selecting "new document". When electronic document 122 is displayed on screen 121, a page grid (also known as a layout grid) 123 is displayed on electronic document 122 based on values set by questions to the user in a dialog box or as preset defaults. Page grid 123 comprises reference lines for arranging various types of objects, such as graphics or characters, within the electronic document. In the example shown here, page grid 123 is formatted as a manuscript form for horizontal writing, but grids with various other formats exist, such as manuscript forms for vertical writing, graph paper format, etc. Page grid 123 usually has a plurality of horizontal lines and a plurality of vertical lines, and also has a plurality of attraction points. Page grid 123's display/nondisplay on electronic document 122 can be controlled by clicking "View" in menu bar 125 and making a pulldown menu appear, and selecting "grid display" or "grid nondisplay" therein. Furthermore, page grid 123's attraction function can be used regardless of display or nondisplay.

In addition, in FIG. 3 frame 20, which is one of the page structural elements (also known as objects), is arranged in electronic document 122. In particular, frame 20 is a text

frame that has a CJK character grid, and has the same format (in this case, as a manuscript

form for horizontal writing) and dimensions as page grid 123. Frame 20 is created by the

user by using a mouse, for example, and clicking on grid creation tool 126b for manuscript

forms for horizontal writing in toolbox 126, and then dragging the mouse along a diagonal

across the desired length of screen 121. The default is to create the grid inside frame 20

identical to page grid 123. Furthermore, immediately after frame 20 is created, frame 20 may

not be coordinated with the position of the page grid, but in this case the position of frame 20

can be coordinated by selecting frame 20 with selection tool 126a, for example, moving the

frame by dragging with the mouse, and selectively attaching (also known as snapping) the

frame to a plurality of attraction points provided on the page grid.

Tool box 126 also has grid creation tool 126c for manuscript forms formatted for

vertical writing, and selection tool 126a for selecting objects displayed on screen 121, and

various other known editing tools. In addition, menu bar 120 also contains functions for

performing common editing tasks such as copying and pasting, and functions for setting

spacing and so forth under "Edit".

FIG. 4 is a flowchart showing the typesetting control procedure in accordance with

one implementation of the invention. This implementation shows the procedure of processing

the typesetting processing of electronic document 122 displayed on screen 121 of a display

device, as shown in FIG. 3 for example. Page grid 123 is set on electronic document 122 in

order to do typesetting processing in electronic document 122, and text frame 20 with

specified dimensions is created on page grid 123 by the user clicking on the horizontal

writing grid tool 126b with the mouse and dragging the tool diagonally across screen 121.

Furthermore, text frame 20's default is to have a grid with the same format and dimensions as

page grid 123. In the example shown in the drawing, the grid of frame 20 has a plurality of

lines extending horizontally, and each line has a plurality of cells. Preferably, each cell has

an essentially square box shape determined by the dimensions of the characters (for example,

their point dimension) that were set when frame 20 was created. In one implementation each

cell can be set to a size that matches CJK character dimensions. In addition, the interline

spacing, that is, spacing between two adjacent lines, is set.

Characters input by the user with a keyboard can be arranged inside frame 20, and

text from a file already created by a word processor, for example, can be arranged inside

frame 20 by flowing the text in. In this case, the characters arranged inside frame 20 in one implementation are first arranged in sequence on the uppermost line from left to right, and then similarly arranged in sequence from left to right on the second line from the top. Then one character is arranged inside each cell within each line, and in this implementation each

5   character is arranged within each cell using a preselected coordination mode. For example, coordination modes may include a top coordination mode, a midpoint coordination mode, a baseline coordination mode, and a bottom coordination mode, etc. based on an "embox". There is also a top coordination mode, a midpoint coordination mode, and a bottom coordination mode, etc. based on an "ICF box".

10   An "embox" is an outer frame around a character, whose vertical and horizontal dimensions are set by the character's dimensions (for example, the character's point dimensions), and essentially corresponds to a so-called "imaginary body". The embox for a 12-point character in a certain font is an essentially square frame whose horizontal dimensions are 12 points and whose vertical dimensions are 12 points, surrounding the

15   character's glyph. FIG. 11 shows one example of an embox. In FIG. 11A, square frame 40 surrounding the outermost sides of the glyph for the character "utsu" is this character's embox. Consequently, an embox can easily be determined if a character's dimensions are known.

       On the other hand, an ICF box (ideographic character face) can be simply described as a character's averaged bounding box. That is, a character has a glyph representing the

20   character portion, and the bounding box is the box surrounding the glyph, formed by the pair of horizontal lines touching the glyph's top and bottom parts and the pair of vertical lines touching positions at the farthest left and farthest right sides of the glyph. An average bounding box obtained by averaging the bounding boxes for one typical character or a

25   plurality of characters in certain fonts (particularly CJK fonts) is called an ICF box. That is, in the example shown in FIG. 11, the embox of CJK character "utsu" in FIG. 11A is outer frame 40, and inner frame 41a, which touches the character and surrounds it, is the bounding box. Then, as shown in FIG. 11B for example, differences a, b, c, and d between embox 40 and bounding box 41a on the top, bottom, left, and right are determined, and the average of

30   these is found, and the averaged bounding box formed inside the embox based on this average value is the ICF box. In another method, FIG. 11C shows embox 40 and bounding

box 41b for the CJK character "naga". Then the differences between embox 40 and bounding box 41 for both characters "utsu" and "naga" are added. The result is averaged to determine the average value for the difference, and the ICF box can be determined based on this average value. In addition, instead of adding all of the top and bottom and left and right differences for a single character and determining the average value, it is also possible to calculate the average value for the left and right differences and the average value for the top and bottom differences separately, and determine an ICF box based on these respective average values. Furthermore, in this case there are instances in which the ICF box is not a perfect square, but by using a character that is essentially square itself, such as "utsu", it is possible to demarcate an ICF box that is essentially square.

As described above, it is possible to determine each character's embox and ICF box, so that the arranging of characters in the individual cells of the grid in frame 20 can be based on the emboxes or ICF boxes. For example, if the grid of frame 20 is created based on emboxes, characters are arranged based on emboxes, while if the grid is created based on ICF boxes characters are arranged based on ICF boxes. When the arrangement is based on emboxes, typically any of the embox's top part (top side), midpoint, baseline, or bottom part (bottom side) are used. On the other hand, if the arrangement is based on ICF boxes, typically any of the ICF boxes' top part (top side), midpoint, or bottom part (bottom side) are used. Furthermore, the example described above is horizontal typesetting processing. In vertical typesetting of course the left and right sides of the emboxes or the left and right sides of the ICF boxes can be used. Therefore, when the coordination mode is based on the top part of an embox, for example, the tops of the emboxes of characters to be arranged are arranged in coordination with the tops of lines (i.e. cells) to be arranged within the grid of frame 20. On the other hand, if the coordination mode is the midpoint of the embox, the midpoints (i.e. centers) of the characters to be arranged are arranged in coordination with the midpoints (centers) of lines (cells) to be arranged within the grid of frame 20.

Next, to explain the inventive processing procedure with reference to FIG. 4, first the process determines hmax, the dimension of the largest character in text consisting of a plurality of characters to be arranged in the current line in frame 20 (step S1). Next, the process decides whether or not the maximum dimension hmax is smaller than the grid dimension (line height or cell dimension) of frame 20 (step S2). If the decision result in step

S2 is negative, an automatic forced grid line spacing procedure based on one implementation of the invention is executed (step S3), while if the result is affirmative the plurality of characters can be accommodated as is in frame 20's grid, so they are arranged in the line corresponding to the current line of frame 20, and the automatic forced grid line spacing procedure is skipped.

Next, the procedure in FIG. 4 moves to step S5, where a decision is made whether or not what is to be coordinated with frame 20's grid is only the first line of one paragraph. If the decision result is affirmative, the procedure immediately ends. On the other hand, if the decision result is negative, the flow moves to the next step, S6, where a decision is made whether or not the current line is the last line of the relevant paragraph. If the current line isn't the last line, the flow branches to step S7, where the next line in the paragraph is set as the current line and the flow returns to step S1. On the other hand, if the decision result in step S6 is affirmative, the procedure ends.

FIGS. 5 and 6 show one implementation of the automatic forced grid line spacing control procedure executed in step S3 in the main procedure of the invention described above. In addition, FIGS. 7 through 10 show how a plurality of characters is arranged in frame 20 when performing processing in accordance with one implementation of the invention. Therefore the flow of processing in accordance with one implementation of the invention will be explained in detail with reference to the flowcharts of FIGS. 4 through 6 and to the schematic diagrams of FIGS. 7 through 10 simultaneously.

First, as shown in FIG. 7A, frame 20 having a manuscript writing grid with a horizontal writing format is prepared in this implementation. Frame 20 has four lines, and each line has six cells 21. Spacing 22 (interline spacing) is set between pairs of adjacent lines. The four boundaries that are formed by a pair of horizontal lines and pair of vertical lines in frame 20 match the boundaries of an internal cell 21. FIG. 7B shows text data (i.e. "computer semiconductor device computer") consisting of a series of characters to be arranged in frame 20 in FIG. 7A; this text data forms one paragraph. This text data is created in advance by a word processor, etc., and stored as a file in a memory device such as a hard disk, etc. As shown in FIG. 11B, this text data is characters of the same CJK font, and with the exception of character 25 (i.e. "tai"), all of the other characters 24 have the same dimension (point dimension) (height h). Only character 25 has a dimension (height H) larger

than the other characters 24. Here, each cell 21 in frame 20 has a coordination mode based on the embox midpoint, so when arranging text data in frame 20 it is assumed that the text is arranged so that the midpoint (center point) of each character's embox is coordinated with the midpoint of the corresponding line (cell 21) of frame 20. Also, it is assumed that the dimensions of small characters 24 in the text data are the same as the dimensions of frame 20's cell 21. In addition, all paragraphs comprising text data in FIG. 7B are arranged in coordination with the grid of frame 20.

Therefore, when arranging the text data of FIG. 7B in frame 20 in FIG. 7A, first, the initial six characters (i.e. "computer") in the text data are selected as the current line, and the dimension hmax of the largest character in this current line is determined (step S1). In this case all of the characters 24 in the current line have the same size (i.e. hmax = h), and this size is the same as the grid (cell) dimension of frame 20, so the flow branches to step S4, and as shown in FIG. 8 these six characters 24 are individually arranged in six cells 21 in the first line of frame 20. In this implementation the entire paragraph is coordinated with the grid of frame 20, so the flow branches to step S7 where the next six characters are set as the current line, and then the flow returns to step S1, and processing begins for the next six characters (i.e. "semiconductor device") in this paragraph.

However, character 25 is included in the current line, and character 25 is larger than the other characters 24, and it is determined that dimension hmax of the largest character in the current line = H (H>h). As described previously, the grid (cell) dimension is the same as the dimension of character 24 (i.e. h), so the current line's hmax (hmax=H) is larger than the grid dimension. Therefore the decision result in step S2 is negative and the flow branches to step S3, and the automatic forced grid line spacing procedure in accordance with the invention is executed.

Next the invention's automatic forced grid line spacing procedure will be explained in detail with reference to both FIG. 5 and FIG. 9.

First, the height H of text ("semiconductor device" in this example) to be arranged in the current line in frame 20's grid is determined to be the height of the embox of the largest character in the text (step S11). Furthermore, this step is the same as step S1 in FIG. 4, so if text height H has been determined before entering the FIG. 5 procedure, step S11 can be skipped. Next, starting position Ys for the current line in frame 20 is determined. As shown

in FIG. 9A, in this implementation frame 20 has at least one line demarcated by the grid structure and extending horizontally, and each line comprises at least one cell 21. A positive coordinate Y is applied whose value increases heading vertically downward, the coordinate Y having a reference value (for example, "0" in the example shown here) which is the top part (top side) of the uppermost line in frame 20. In addition, in the implementation shown here frame 20 has a spacing (interline spacing) of a specified dimension adjacent to the bottom of the line for text arrangement (except for the bottommost line). Furthermore, the size of cell 21 and the size of the spacing amount can be set by the user to an optional size when creating frame 20 and at an optional time thereafter.

As described previously, starting position Ys for the current line is determined as the bottom position of the previous line's spacing (in this example, the uppermost line in frame 20), as shown in step S12 of FIG. 9A. Next, as shown in the flowchart in FIG. 5, first rectangle 30 (L, R, Ys, Ys + 1) is determined from the current line in frame 20 (step S13). That is, the top side of first rectangle 30 is a horizontal line at the position of Ys, and the bottom side is a horizontal line at the position of Ys + H (text height for the current line), and the left side is vertical line L (part of the left side of frame 20 in this example) passing through the edge of the left side of the current line in frame 20, and the right side is vertical line R (part of the right side of frame 20 in this example) passing through the edge of the right side of the current line in frame 20.

Next, a decision is made as to whether or not a grid coordination mode is selected (step S14), and if the decision result is negative the flow branches to step S22 and text is arranged in first rectangle 30 shown in FIG. 9A (step S22). On the other hand, if the decision result in step S14 is affirmative, the selected grid coordination mode GAV is determined (step S15). Furthermore, grid coordination mode GAV regulates the sort of coordination reference used to arrange characters in the grid of frame 20 when arranging characters in frame 20. For example, it regulates whether the top, center, baseline, or bottom of a character embox (or the top, center, or bottom of an ICF box) is coordinated with the grid reference line of frame 20 in arranging characters. Also, in this implementation the grid coordination mode GAV is selected in paragraph units, and the selected GAV is stored at a specified storage location in memory. Therefore, when the user selects a grid coordination mode GAV for a specified paragraph, that GAV is effective for all lines in the paragraph. Furthermore, this

implementation assumes that grid coordination mode GAV is selected as the embox center (in different implementations, of course, a different GAV can be selected).

Next, based on the selected grid coordination mode GAV, point Ya in first rectangle 30 is determined. In this implementation, GAV is embox center, so in this case point Ya is the center of first rectangle 30—that is, the center position Ya in the vertical direction between the top side Ys and bottom side Ys + H of the first rectangle 30—and is determined as Ys + H/2. On the other hand, when GAV is embox top, for example, point Ya is set as the first rectangle's top part Ys (that is, in this case Ya = Ys).

Next, a decision is made whether the grid of frame 20 is active (activated status) or not (step S17). If a CJK grid based on cell 22 in frame 20 is not active, or if a grid does not exist, text cannot be coordinated with the grid of frame 20, so the flow branches to baseline processing (step S19), and characters are arranged in frame 20 using a baseline grid (a grid consisting of a plurality of horizontal lines arranged at fixed intervals in order to coordinate characters referenced on a baseline) set in frame 20 by default in accordance with the procedure shown in FIG. 6. That is, in this case, as shown in the FIG. 6 flowchart, the next grid line's (baseline) position Yc is determined in the baseline grid (step S30), and then positional difference DY in the Y direction is determined as DY = Yc – Ya (step S31). Subsequently the processing returns to the flowchart in FIG. 5.

On the other hand, if the frame grid is active, i.e. if a CJK grid based on cell 22 in frame 20 is active, as shown in FIG. 9B second rectangle 31 surrounding the current line and at least one following line (one line in this implementation) is determined, and then point Yb is determined for this second rectangle 31 again based on GAV. That is, in this implementation GAV is the embox center, so second rectangle 31's center position—that is, the point at the position in the center between second rectangle 31's top side Ys and bottom side (the third line in frame 20, i.e. the bottom side of the line adjacent to the bottom of the current line)—is determined as point Yb. Furthermore, in this implementation, when text height H is larger than the grid's line height, the second rectangle is demarcated by adding one other line to the current line, but of course the number of lines to be added to the current line can be set to an optional number in accordance with the forced grid line spacing mode desired by the user (for example, taking two lines, three lines, etc.). In addition, this second rectangle is imaginary, so demarcating the second rectangle itself is not always important in

this procedure. If it is possible to obtain the length in the Y direction from Ys according to a desired forced grid line spacing mode and determine point Yb according to the GAV selected for that length, that may be sufficient. For example, when GAV is the embox top, position Ys becomes the reference line for arranging characters, so determining second rectangle 31 is not particularly necessary in order to determine point Yb. Nevertheless, in this case too it is necessary to demarcate second rectangle 31 in order to determine the scope of the desired forced grid line spacing in order to arrange the next line in frame 20, of course.

Next, as shown in FIG. 9B, positional difference DY in the Y direction is determined as DY = Yb − Ya (step S20). Next, as shown in FIG. 9C, first rectangle 30 is moved downward by exactly DY. Next, first rectangle 30' moved to the new position becomes the current line, and one line's worth of characters is arranged therein (step S22). In this way a character string having a character 25 that is larger than the grid dimension of frame 20 is positioned and arranged in the center of a space demarcated by two lines (and the spacing region between them) in a two forced grid line spacing configuration in this implementation.

In addition, the remaining characters 24 in the paragraph are all the same size as frame 20's grid dimension, so these remaining characters 24 are sequentially arranged in the last line of frame 20. FIG. 10 shows the final result.

As described above, when the maximum dimension of a plurality of characters being arranged in the current line of frame 20 exceeds the grid dimension, the invention automatically executes forced grid line spacing for exactly the number of lines preselected or set, and makes it possible to arrange characters in a space demarcated by a plurality of lines with forced grid line spacing and a specified coordination mode. Therefore it is not necessary for the user to perform manual operations or complicated setting operations in order to execute forced grid line spacing, and typesetting operations can be performed extremely efficiently.

Potential advantages of the invention in accordance with one or more of the implementations described above may include the following. The invention may eliminate the conventional setting technique described above and may provide a typesetting technology provided with a forced grid line spacing control function that can automatically and swiftly perform forced grid line spacing even of text consisting of a plurality of characters with differing fonts or dimensions. The invention may provide a forced grid line spacing

technology such that when the height of text consisting of a plurality of characters to be arranged within one line of a grid displayed on a display device is larger than the grid dimension, a plurality of lines can be selected and an arrangement region can be demarcated, and the reference point of a plurality of characters can be coordinated with the reference point of the arrangement region and the characters can be arranged in the arrangement region according to a preselected coordination mode. The invention may provide a DTP system that provides an improved forced grid line spacing control function. The invention may provide a forced grid line spacing control method that can be executed by a computer. The invention may provide a computer-readable recording medium storing an improved forced grid line spacing control program.

A number of implementations of the invention have been described. Nevertheless, it will be understood that various modifications may be made without departing from the spirit and scope of the invention. Accordingly, other implementations are within the scope of the following claims.